

Screen-Space Curvature for Production-Quality Rendering and Compositing

Nicolas Mellado¹ * Pascal Barla¹ Gael Guennebaud¹ Patrick Reuter¹ Gregory Duquesne²
1 - Inria - Univ. Bordeaux - IOGS - CNRS 2 - Luxology

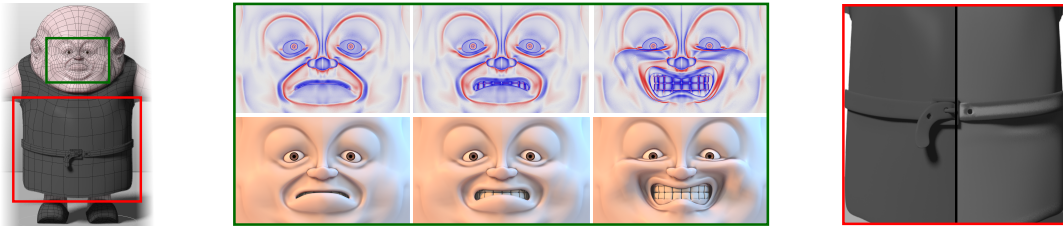


Figure 1: A subdivision surface character (courtesy of Peter Eriksson) is animated and rendered using Modo©. Three poses are shown in the middle: our screen-space mean curvature (top: concave regions in red, convex regions in blue) is used to modulate a basic shading (bottom-left) yielding an exaggerated appearance (bottom-right). Right: another shading modulation based on our offset τ .

1 Introduction

Surface curvature is a measure commonly employed in Computer Graphics for a vast range of applications: for modeling purposes of course, but also to drive texture generation, or to produce exaggerated or stylized shading results (see Figure 1).

Curvature is a differential property expressed at each surface point. When working with meshes, it can be efficiently computed per vertex using GPU-based techniques [Griffin et al. 2011]. However, this approach becomes impractical when dealing with production quality scenes that involve various representations (e.g., subdivision surfaces, implicit and procedural geometry, displacement and bump maps) and/or deformable objects. This severely limits its use in real-time editing (as in sculpting) or for shading animated scenes.

2 Screen-Space Curvature

Our approach may be applied either in ray-tracing or compositing contexts. For each sample (ray or pixel), we start by collecting neighbor samples in screen-space that are closer than a threshold distance t . We also reject neighbors that are farther than a threshold relative depth d , to avoid collecting samples across occluding contours. We then recover a normal vector and a 3D position for each neighbor sample. The resulting *local point cloud* is fit with an algebraic sphere where each sample is weighted by a compactly-supported function parametrized by t .

Our fitting is performed using the method of Mellado et al. [2012], which has the benefit of characterizing the resulting sphere with geometrically-meaningful parameters. This provides us not only with a mean curvature estimate κ , but also with an offset τ that identifies local relief and a smooth reconstructed normal vector η . With this approach, t controls the scale of the local reconstruction. As shown in Figure 2, our approach produces an accurate approximation of object-space mean curvature, which is due to the local nature of differential properties. There is a notable difference though: farther objects exhibit less details than closer ones with the screen-space approach, which provides for automatic simplification and reduced aliasing artifacts.

An alternative to our screen-space curvature would be to apply 2D derivative filters on normals [Vergne et al. 2009]. However, this

approach neglects 3D positions, and raises issues nearby occluding contours, whereas our fitting remains well-defined.

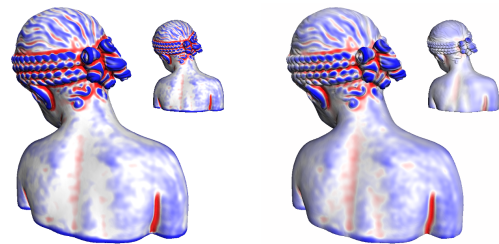


Figure 2: Comparison between mean curvature estimated in object- (left) and screen-space (right), shown at 2 different scales. 3D model courtesy of AIM@SHAPE Library.

3 Applications

We have implemented SSC in Modo© on top of its ray-tracing engine, and in CUDA for post-processing, as shown in the accompanying video. An example animation is depicted in Figure 1, where we visualize our mean curvature estimate κ with a color code, along with a simple curvature-based shading result. Our method is applied at each frame with a negligible performance overhead compared to rendering (for both pixel- and ray-based implementations), and it exhibits natural temporal coherence.

In future work, we plan to use SSC during sculpting to enhance small (yet important) surface details, and for non-photorealistic rendering to guide stylization. In both cases, we believe that not only κ but also τ and η will prove to be valuable control parameters.

References

- GRIFFIN, W., WANG, Y., BERRIOS, D., AND OLANO, M. 2011. GPU curvature estimation on deformable meshes. In *ACM 13D '11*, 159–166.
- MELLADO, N., GUENNEBAUD, G., BARLA, P., REUTER, P., AND SCHLICK, C. 2012. Growing least squares for the analysis of manifolds in scale-space. *CGF 31*, 5, 1691–1701.
- VERGNE, R., PACANOWSKI, R., BARLA, P., GRANIER, X., AND SCHLICK, C. 2009. Light warping for enhanced surface depiction. *ACM Trans. on Graph.* 28, 3.

*e-mail: nicolas.mellado@inria.fr